**class** Student {

**private** **int** rno;

**private** String name;

**public** Student(**int** r, String n) {

rno = r;

name = n;

}

@Override

**public** String toString() {

**return** rno + " " + name;

}

}

**public** **class** Demo

{

**public** **static** **void** main(String[] args)

{

Student s = **new** Student(101, "Susan Bones");

System.***out***.println("Los atributos del Objeto s de la clase Student serán:");

System.***out***.println(s);

System.***out***.println( "Valor entero: " + s.hashCode());

System.***out***.println("Valor Hexadecimal: " + Integer.*toHexString*(s.hashCode()));

}

}

**//Explicación: La sentencia** System.***out***.println(s); **por estar** **más arriba**

@Override

**public** String toString() {

**return** rno + " " + name;

}

**Va a ser llamado este método que retornará el formato en que serán impresos**

**los atributos del Objeto s.**

**El método toString() pertenece al Java, pero con el @Override se le está diciendo que tiene que devolver cuando se quiera imprimir un Objeto. En este caso** System.***out***.println(s);

**Va a funcionar el toString() escrito bajo el @Override y generará la salida:**

101 Susan Bones

Pruebe el Programa para verificar lo explicado